

# What is WordPress Nonce? (Preventing CSRF Attack)

BY HENNER SETYONO / ON 14 AUG 2020

#GENERAL



Let's say you installed a **Forum plugin** that can create a new thread, manage your profile, change password, etc.

Then someone malicious came in and post a link that will trigger password change to "123456". The link might say "Won \$10.000" or something attractive so more people got tricked by it.

Now that person have access to the victim's account.

This kind of attack is called **CSRF** (Cross Site Request Forgery) where a 3rd party sends a fake request. The damage can be devastating.

Is the above scenario possible? Yes, unless the plugin author added a prevention measure using **Nonce**.

## What is Nonce?

Nonce is a **randomly-generated string** attached to a URL or Form to verify that an action is done by the user him/herself.

The string is one-time use and different for each user. That means it's impossible to fake. If the nonce is not valid, we can reject the request.

Here's a diagram showing 2 requests, 1 from the user, and 1 from a hacker that got rejected:

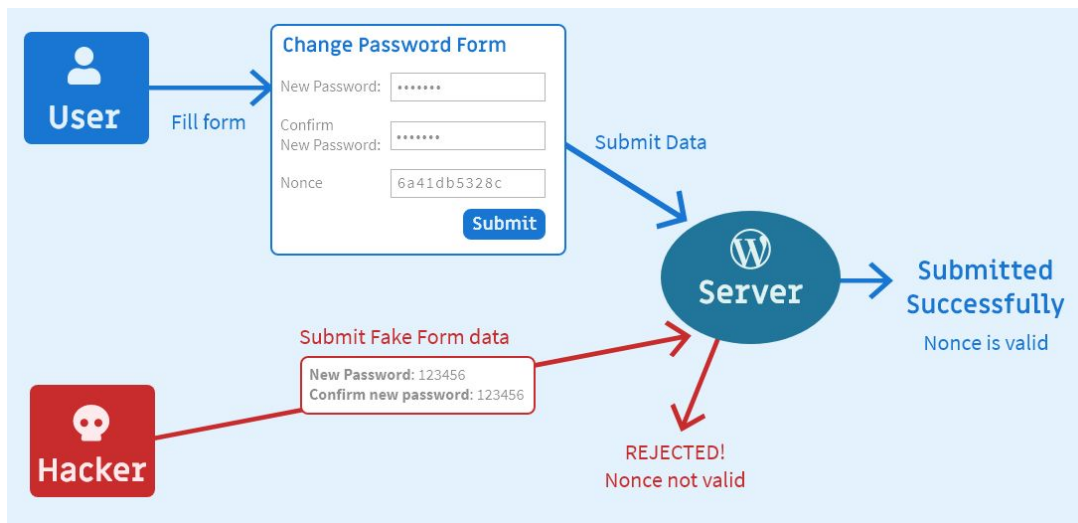


Diagram showing how nonce works.

Hacker that posts fake data will be rejected because nonce is empty or not valid.

## Is My Website Vulnerable to CSRF?

First, identify which of your link or form **could cause negative effect** on your site.

Something public like Contact Form is fine without nonce. What's the worst thing that could happen anyway? Get spammed? Nonce won't solve that.

### Found Potentially harmful Form?

For example, we have the Password Change form in WooCommerce account page.

Open Web Inspector (F12) and check whether the form has a hidden field containing the nonce or not.



WooCommerce's password change form has nonce field

Yes, it has a nonce field. But to be sure, try removing that nonce field and submit the form.

Does it still change the password? In the case of Woocommerce, it fails (which is the correct behavior).

## Found Potentially harmful Link?

For example, we have WooCommerce Marketplace plugin. Users can post and delete their own products.

That delete button seems dangerous right?

Open web inspector and check out the link. If it has a nonce then it's safe:

```
https://myshop.com/delete-product/1042?_wpnonce=75ad822113
```

But just to be sure, copy the link, and remove the nonce. So you have:

```
https://myshop.com/delete-product/1042
```

Then paste it in a new tab. See if your product got deleted. If it's deleted, your site is vulnerable to CSRF.

## How to Implement Nonce?

This is useful if you want to create your own form or plugin.

### GENERATE NONCE

There are 3 ways to generate a nonce:

- `wp_create_nonce()` – Create a plain nonce, use it anyway you like.
- `wp_nonce_url()` – Appending a `_wpnonce` parameter to a URL.
- `wp_nonce_field()` – Echoing a hidden field containing nonce.

Example:

```
$nonce = wp_create_nonce( 'add_product' );  
// zxcvbn678  
  
$url = 'https://myshop.com/';  
$nonce_url = wp_nonce_url( $url, 'delete_product' );  
// https://myshop.com/_wpnonce=abcdef123  
  
wp_nonce_field( 'change_password' );  
// <input type="hidden" id="_wpnonce" name="_wpnonce" value="qwerty123" />
```

PHP

## VERIFY NONCE

Use `wp_verify_nonce()` by passing in the nonce and its name.

For example, the code below handles form submission:

```
$nonce = $_POST['_wponce'];

if( wp_verify_nonce( $nonce, 'change_password' ) ) {
    // success!, process the password change
    // ...
} else {
    return false; // abort it
}
```

PHP

• • •

## Conclusion

CSRF is the **most common vulnerabilities** found in WordPress plugins.

It is important to be cautious when choosing a plugin, especially when it has low popularity. A small amount of users means it takes longer for someone to found the loophole.

Having said that, popular plugins are not always 100% safe. But when a potential threat is found, usually it is quickly patched. So it's important to always **keep your plugins up-to-date**.

That's all, I hope you found this article useful. If you have any question, please post it in the comment below 😊

# php

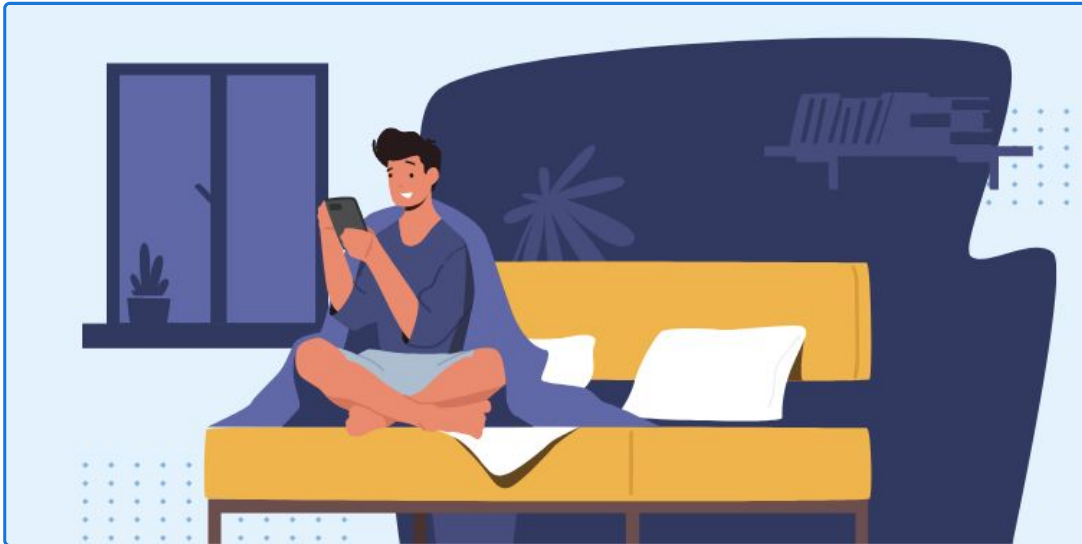
# security



Written by  
**Henner Setyono**

A web developer who mainly build custom theme for WordPress since 2013. Young enough to never had to deal with using Table as layout.

## Related Posts



### Quick and Easy Dark Mode without Plugin

On 2 Aug 2022



### Our WordPress Workflow with Github Action

On 28 Feb 2022 / 2 Comments



## 7 Tips to Avoid Spaghetti Code in WordPress (and PHP)







On 31 Jul 2020 / 2 Comments

### Leave a Reply

Name \*

Email \*

Website

**B** *I*      Block 

- ☐ Save my name, email, and website in this browser for the next time I comment.
- ☐ Notify me of follow-up comments by email.

Post Comment



Design by [Pixel Studio](#)

## About wpTips.dev

wpTips is a non-profit blog focused on **advanced** WordPress tutorial in a concise and beautiful way.

We want to show the world the full potential of our beloved CMS.

## Join Our Community

We are a team of enthusiasts who enjoy sharing our knowledge.

Let us know at [hello@wptips.dev](mailto:hello@wptips.dev) if you are interested to write some tips!

Copyright © 2022 WPTips.dev